

Computer algebra in gravity*

Christian Heinicke[†]

Friedrich W. Hehl[‡]

Institute for Theoretical Physics

University of Cologne

50923 Köln, Germany

February 7, 2008

Abstract

We survey the application of computer algebra in the context of gravitational theories. After some general remarks, we show of how to check the second Bianchi-identity by means of the Reduce package Excalc. Subsequently we list some computer algebra systems and packages relevant to applications in gravitational physics. We conclude by presenting a couple of typical examples.

Contents

1. Introduction
2. Riemannian curvature in tensor and exterior calculus
3. Abstract and component CA systems
4. General versus special purpose systems
5. Applications
6. References

*Updated version of an article to appear in: *Computer Algebra Handbook*. J. Grabmeier, E. Kaltofen, V. Weispfennig, editors (Springer, Berlin 2001/2002).

[†]Email: chh@thp.uni-koeln.de

[‡]Email: hehl@thp.uni-koeln.de

1 Introduction

Einstein's gravitational theory, *general relativity* (GR), is the valid theory for describing gravitational effects. In the search for making GR compatible with quantum theory and/or unifying it with the other interactions of nature (strong, electro-weak, superweak,...), different schemes have been developed, like the gauge approach to gravity, including supergravity and metric-affine gravity, higher-dimensional Kaluza-Klein type models, string models, but also more conventional Hamiltonian (canonical) or Feynman quantization schemes or, more far-fetched, models based, e.g., on noncommutative spacetime geometries.

The computer algebra (CA) programs applied in GR can be and partially have been extended to these more general frameworks. Still, it is probably true that most CA programs in gravity are applied in the context of GR followed by those for evaluating gravity-based Feynman integrals and for executing computations in the framework of gauge models encompassing non-Riemannian spacetimes. In this note we mainly concentrate on GR and will give a couple of examples.

Detailed overviews of computer algebra in GR are given, e.g., by Brans [3], Hartley [12], Lake [17], and MacCallum [21], [20].

2 Riemannian curvature in tensor and exterior calculus

In GR and gravity, computer algebra was used as soon as it became available. The reason for this is that for solving standard problems it is required to manipulate a large number of terms and equations. We will clarify this by an example. A generic problem in gravity is to calculate the Ricci tensor from a given metric. A general form of a spacetime metric g in four dimensions is given by 10 independent functions $g_{ij} = g_{ji}$ of the coordinates (x^0, x^1, x^2, x^3) :

$$g = \sum_{i,j=0}^3 g_{ij}(x^0, x^1, x^2, x^3) dx^i \otimes dx^j . \quad (1)$$

The so-called Christoffel symbols are determined from the functions g_{ij} by means of the following equations:

$$\Gamma_{ij}{}^k = \frac{1}{2} \sum_{m=0}^3 g^{km} \left(\frac{\partial g_{jm}}{\partial x^i} + \frac{\partial g_{mi}}{\partial x^j} - \frac{\partial g_{ij}}{\partial x^m} \right) = \Gamma_{ji}{}^k . \quad (2)$$

Here g^{km} denotes the matrix reciprocal to g_{ij} . Since i, j, k are running from 0 to 3, the $\Gamma_{ij}{}^k$ represent 64 functions. Because of the symmetry in i, j , only 40 are independent. In the conventions of Schouten [31], the Riemannian curvature tensor is derived from the Christoffel symbols in the following way:

$$R_{ijk}{}^l = \frac{\partial \Gamma_{jk}{}^l}{\partial x^i} - \frac{\partial \Gamma_{ik}{}^l}{\partial x^j} + \sum_{m=0}^3 \left(\Gamma_{im}{}^l \Gamma_{jk}{}^m - \Gamma_{jm}{}^l \Gamma_{ik}{}^m \right) . \quad (3)$$

Eventually we find the Ricci tensor as

$$\text{Ric}_{jk} = \sum_{i=0}^3 R_{ijk}{}^i . \quad (4)$$

Now one can easily estimate that the number of terms in each of the components Ric_{ij} may be very large. In [10] it is shown that in the general case this number is in the order of 10 000 for each of the components. Thus, only in simple cases can these calculations be done conveniently by hand.

It would be most desirable to have a computer algebra system which allows to enter mathematical expressions in an analogous way as one would write them down on paper. That is, defining objects with abstract properties, doing calculations, and assigning explicit values to these objects should be possible in a natural way. Such kind of systems already exist. We illustrate this by an example.

In terms of Cartan's calculus of exterior differential forms, Eq.(3) can be displayed very compactly as

$$R_\alpha{}^\beta = d\Gamma_\alpha{}^\beta - \Gamma_\alpha{}^\gamma \wedge \Gamma_\gamma{}^\beta . \quad (5)$$

Here, $R_\alpha{}^\beta$ is the curvature 2-form and $\Gamma_\alpha{}^\beta$ the connection 1-form. The summation convention is assumed, i.e. summation is understood over the repeated index γ . Let us check the *Bianchi identity*

$$dR_\alpha{}^\beta - \Gamma_\alpha{}^\gamma \wedge R_\gamma{}^\beta + \Gamma_\gamma{}^\beta \wedge R_\alpha{}^\gamma = 0 \quad (6)$$

on the computer. This can be deduced from Eq.(3) and the properties of the exterior product and the exterior derivative. In Excalc, a Reduce package for exterior calculus, we first have to declare that all indices run from 0 to 3. Then we declare Γ_α^β to be a 1-form and R_α^β a 2-form.

```
indexrange 0,1,2,3;
```

```
pform gamma(a,b) = 1 , curv(a,b) = 2;
```

Eq.(5) and the left hand side of Eq.(6) can almost literally be translated,

```
curv(-a,b) := d gamma(-a,b) - gamma(-a,c) ^ gamma(-c,b);
```

and

```
d curv(-a,b) - gamma(-a,c) ^ curv(-c,b) + gamma(-c,b) ^ curv(-a,c) ;
```

The negative sign in front of an index indicates that it is a subscript whereas a superscript is denoted by a positive (or no) sign. The last command yields zero, i.e. the Bianchi identity is confirmed. In [28, p.234] it is shown how to handle this in the Mathematica package MathTensor.

3 Abstract and component CA systems

Packages (or systems) capable of performing symbolic calculations, as with the Bianchi identity above, are called *abstract* or *indicial* calculus systems [12]. They are necessary if one wants to investigate general properties of objects. So called *component* calculus systems are designed to calculate the components of unknown quantities from known ones. In our example, it was not necessary to introduce a basis or a metric in order to define Γ_α^β and R_α^β . In a typical component system, one first would have to enter the components of a metric. Then, by means of build-in routines, the components of the connection and curvature could be calculated. Both, abstract as well as component systems, allow to define new objects by means of various mathematical operations like products or derivatives, e.g.. The difference here is that in the case of a component system it is always necessary to assume a specific basis and/or metric. In turn, there are abstract calculus systems which do not support computations of explicit values of components. Some packages, like MathTensor or Excalc, for instance, allow both, abstract and component calculations (see Table 1). In our Excalc example, we could assign explicit expressions to the components of Γ_α^β . Then we would find the components of R_α^β by calling `curv(-a,b)`.

4 General versus special purpose systems

Today it seems that most people use relativity packages of *general purpose systems* like Maple, Mathematica, Reduce, Macsyma and/or Derive. These programs offer very user-friendly front-ends and, moreover, a wealth of useful facilities, like simplification routines, programs for solving algebraic or differential equations exactly or numerically, Tex and Fortran interfaces, etc..

Special purpose systems are, as the name suggests, specialized to handle only a specific class of problems. Therefore the set of instructions is usually very limited and the programming of these systems requires normally much more effort than is the case with general purpose systems. However, these systems are rather compact, very fast, and may sometimes be the only available facility to solve a problem. In the case of calculating Feynman diagrams, e.g., special purpose systems like Schoonship or Form are often used, but there are also packages available for Mathematica and Reduce. A fairly widely used special purpose system for tensor calculus and general relativity is Sheep. For general relativity formulated in terms of orthonormal frames, the special purpose system Ortocartan has been recently updated [16].

System	Component	Abstract
Macsyma [23]	CTensor	ATENSOR ITENSOR CARTAN
Maple[24]	tensor cartan NPspinor debever oframe GRTensorII [9] Riemann [25]	diffforms forms
Mathematica[26]	Cartan [35] TTC [1] MathTensor [28], [4]	EinS [15] Ricci [18] MathTensor DifferentialForms
Reduce [30]	EXCALC Chap. REDTEN[11] GRG GRG _{EC}	EXCALC RICCIR [14] GRGlib [22]
Sheep [19], [21], [33]	CORD FRAME	STENSOR

5 Applications

Today many authors use computer algebra in order to obtain or confirm their results *without* mentioning it explicitly. In the following we present some articles which explicitly illustrate the applications of computer algebra in gravity.

It was already stated that quantities like the *Ricci tensor* can reach an enormous size. Thus, for most applications (classification, numerics), these objects have to be put on the computer. The use of computer algebra has the advantage that there is no need to enter very large expressions for the curvature, e.g., but a comparably small input program which calculates these. Moreover, one can use the possibility to transmit programs or results by means of computer networks, cf. the closing remark in [7].

A standard application of computer algebra in GR is the *classification of exact solutions*. These are necessarily found in special coordinates. However, it is desirable to characterize the corresponding spacetimes in a coordinate independent way. This includes determination of Petrov and Segre types, calculations of curvature invariants (the trace of the Ricci tensor of Eq.(4), for instance), the maximal isotropy group, etc.. The appropriate algorithms involve an enormous amount of calculational work. Programs for the widely used Petrov classification are available for most computer algebra systems. By means of the Sheep-package Classi [21], it was possible to create a searchable online-databank which includes nearly 200 exact solutions and their properties and which is still growing [32].

To find out whether two solutions which look different are not just the same solution in different coordinates, one has to solve the so-called *equivalence problem*. This involves differentiation of the curvature tensor up to the seventh order. In [21] appropriate programs are presented for Sheep.

Computer algebra is also very useful for finding *new solutions* of the Einstein equation. For vacuum, it reads $\text{Ric}_{ij} = 0$. As we can recognize from Eq.(4) together with Eqs.(2) and (3), it represents a system of ten second order quasi-linear partial differential equations for the g_{ij} which are obviously very difficult to solve. A simple example is given in [8] where the Schwarzschild solution is derived from a spherical symmetric line element by using the Reduce package Excalc. In [36] it is illustrated how to use the Reduce-based system GRG_{EC} for searching for solutions of the Einstein-Maxwell equations. In [39] the Reduce package Classym was used to derive the Killing vector and Killing tensor equations and their integrability con-

ditions from a general form of a metric. Subsequently the Reduce package Crack has been used for solving these equations.

In [34] it is shown how to construct solutions of *metric-affine* gravity from solutions of the Einstein-Maxwell equations under heavy use of the computer algebra system Reduce.

A further application is the derivation of a field equation from an action principle by means of *variational calculations*. A simple example is given in [28, p.303] where the Einstein vacuum equation is rederived. In [37], the gravitational field equation of a unified field theory of Pawłowski and Rączka were derived (and corrected) from the corresponding Lagrangian by means of MathTensor.

In [38] an example is given of how to use the special purpose system Form to calculate *Feynman diagrams* in the context of quantum gravity.

Another feature of many computer algebra systems is a *Fortran interface* which converts equations into Fortran readable form. This helps to develop programs for numeric calculations, as illustrated in [2].

In [6] it is outlined of how to apply the computer algebra system GRTensorII to second-order black hole perturbations.

References

- [1] A. Balfagón, P. Castellví and X. Jaén: *TTC: Symbolic Tensor Calculus with Index*. Available at: <http://baldufa.upc.es/ttc/>
- [2] S. R. Brandt and E. Seidel: *The evolution of distorted rotating black holes I: Methods and tests*. Phys.Rev. **D 52** 856–69 (1995).
Also available at: <http://arXiv.org/abs/gr-qc/9412072>
- [3] C. H. Brans: *Computer algebra and general relativity*, in J. Fleischer, J. Grabmeier, F. W. Hehl, and W. Küchlin, editors, *Computer Algebra in Science and Engineering*, pages 183–195. World Scientific, Singapore, 1995.
- [4] S. Christensen: *MathTensor online documentation*.
Available at: <http://smc.vnet.net/MathSolutions.html>
- [5] *Computer algebra information network*.
Available at: <http://www.can.nl/>

- [6] G. Davies: *Second-order black hole perturbations: A computer algebra approach, I – The Schwarzschild spacetime* (1998).
Available at: <http://xxx.lanl.gov/abs/gr-qc/9810056>
- [7] F. J. Ernst, A. D. Garcia, and I. Hauser, *Journal Math. Phys.* **28** 2155–2161 (1987).
- [8] F. Ghergu and D. Vulcanov. *Use of computer facilities in teaching general relativity* (1998).
Available at: <http://xxx.lanl.gov/abs/physics/9812004>
- [9] *GRTensorII: Online documentation and information.*
Available at: <http://grtensor.phy.queensu.ca/>
- [10] *GRTensorII demonstration page—general relativity & geometry.* Available at: <http://grtensor.phy.queensu.ca/NewDemo/demo.html#classic>
- [11] J. F. Harper and C. C. Dyer: *Tensor Algebra with REDTEN.* Available at: <http://www.scar.utoronto.ca/~harper/redten/root.html>
- [12] D. Hartley: *Overview of computer algebra in relativity* (1996), in [13], pp. 173–191.
- [13] F. W. Hehl, R. A. Puntigam, and H. Ruder, editors: *Relativity and Scientific Computing*, Springer-Verlag, Berlin, 1996.
- [14] J. Kadlecik: *RICCIR: Ricci calculus package in REDUCE* (1996).
Available at:
<http://www.kfki.hu/cnc/szhkpub/riccir/riccir.html>
- [15] S. A. Klioner: *EinS: A Mathematica package for calculations with indexed objects. Information.* Available at:
<http://rcswww.urz.tu-dresden.de/~klioner/eins.html#abstract>
- [16] A. Krasinski: *The newest release of the ortocartan set of programs for algebraic calculations in relativity.* *Gen. Rel. Grav.* **33** 145–161 (2001).
- [17] K. Lake: *GR 15 proceedings A5(ii) computer methods in GR: Algebraic computing.* Available at: <http://xxx.lanl.gov/abs/gr-qc/9803072>

- [18] J. M. Lee: *Ricci: A Mathematica package for doing tensor calculations in differential geometry. Documentation and source code.*
<http://www.math.washington.edu/~lee/Ricci/>
- [19] M. A. H. MacCallum: *Sheep: Information and source code.*
<http://www.maths.qmw.ac.uk/hyperspace/#ftp>
- [20] M. A. H. MacCallum: *Computer algebra and applications in relativity and gravity*, in A. Macias, T. Matos, O. Obregon, and H. Quevedo, editors, *Recent Developments in Gravitation and Mathematical Physics: Proceedings of the First Mexican School on Gravitation and Mathematical Physics*. World Scientific, Singapore, 1996.
- [21] M. A. H. MacCallum and J. E. F. Skea: Sheep: A computer algebra system for general relativity (1994), in [29], pp. 1–172.
- [22] J. D. McCrea, *REDUCE in General Relativity and Poincaré Gauge Theory* (1994), in [29], pp. 173–263. See the library
<ftp://ftp.maths.qmw.ac.uk/pub/grlib/>
- [23] Macsyma product information. <http://www.macsyma.com/>
- [24] Maple product information. <http://www.maplesoft.com/>
- [25] Maple package Riemann: Documentation and source code.
<http://www.astro.queensu.ca/~portugal/Riemann.html>
- [26] Mathematica product information. <http://www.wri.com/>
- [27] Mathematica package Cartan: Product information and excerpt of the Cartan manual. Available at:
<http://www.universitetsforlaget.no/books/en/cartan/>
- [28] L. Parker and S. M. Christensen: *MathTensor: A System for Doing Tensor Analysis by Computer*. Addison-Wesley, Redwood City, 1994.
- [29] M. J. Rebouças, W. L. Roque, editors, *Algebraic Computing in General Relativity (Lecture Notes from the First Brazilian School on Computer Algebra, vol. 2)*. Oxford University Press, Oxford, 1994.
- [30] Reduce online documentation and information.
<http://www.uni-koeln.de/REDUCE/index.html>

- [31] J. A. Schouten, *Tensor Analysis for Physicists*. 2nd ed. reprinted. Dover, Mineola, New York, 1989.
- [32] J. E. F. Skea et al.: *On-line invariant classification database*.
Available at: <http://www.astro.queensu.ca/~jimsk/>
- [33] J. E. F. Skea: *Applications of SHEEP* (1994). Available at:
<http://www.can.nl/SystemsOverview/Special/Tensoranalysis/SHEEP/shpdrv.ps.gz>
- [34] J. Socorro, A. Macias, and F. W. Hehl: *Computer algebra in gravity: Reduce-Excalc programs for (non-)riemannian spacetimes*. I, *Comp. Phys. Comm.* **115** 264-283 (1998). Also available at:
<http://xxx.lanl.gov/abs/gr-qc/9804068>
- [35] H. H. Soleng: *Tensors in Physics*. Scandinavian University Press, Oslo, 1996.
- [36] S. I. Tertychniy: *Searching for electrovac solutions to Einstein-Maxwell equations with the help of computer algebra system GRG_{EC}* (1998). Available at: <http://xxx.lanl.gov/abs/gr-qc/9810057>
- [37] E. Tsantilis, R. A. Puntigam, and F. W. Hehl: *A quadratic curvature Lagrangian of Pawłowski and Rączka : A finger exercise with Mathtensor* (1996), in [13], pages 231–240.
Also available at: <http://arXiv.org/abs/gr-qc/9601002>
- [38] A. E. M. van de Ven: *Two-loop quantum gravity with the computer algebra program form* (1996), in [13] , pages 192–209.
- [39] T. Wolf: *The program crack for solving pdes in general relativity* (1996), in [13], pages 241–258.